NEAC-2203 による HARP 5020 ソースプログラムのエラーチェック*

小玉忠弘**　鹿野洋治**

1. はじめに

現在，学内の共同利用の計算機は完全にオーバーフローの状態にあり，東京大学大形計算機センターを能率よく積極的に利用しなければならない実状である．

しかし新しいプログラムを作った場合，つまらない文法ミスとかパンチミスを犯しやすく，一度で通すことは非常に困難であり，地方のユーザーに関しては大形計算機センターのターンアラウンドタイムは 5～6 日に及ぶ．計算結果を今やおそしと待ちうけていて，得られたものはエラーメッセージということは誰でもよく経験するものである．

そこでパンチミス，文法ミスを事前に発見し，時間のロスを省き，計算の能率化をはかるために NEAC-2203 を使って HARP 5020 ソースプログラムのチェックプログラムを作成した．

先にものべたように，この計算機は非常に混んでいるため，このチェックプログラムの能率を高めることが初めから望まれている．そこで利用者の計算結果のプログラムエラーを種々検討した結果，エラー総数の 8 割程度までチェックできれば実用になると判断し，以下この線に沿って行なわれている．

2. 使用機器

本体（NEAC-2203）

コア	800 語（内，インデックス指定可能は 400 語）
ドラム	2,000 語
外部ドラム	10,000 語
ラインプリンター	1 台
紙テープ読取機	1 台
カード→テープ変換機	1 台（オフライン）

* Error Check for HARP 5020 Source Program, by Tadahiro Kodama and Yoji Shikano (the Faculty of Engineering, Nagoya University)

** 名古屋大学工学部総合計算室

3. チェックプログラム

処理はデック単位で行ない，処理の対象とする文はハードウェア IF 文，DEBUG 文それに MT に関する文を除く全ての文とした．

プログラムはセンターへ送るままの形でチェックするのが望ましいが，当計算機室のカード読取せん孔機はハードウェア上種々の制限があり，この目的には実際上使用できないため，カードをオフラインで紙テープに変換する装置であらかじめ紙テープにしておく．

処理プログラムは記憶容量からくる制限と処理速度を上げるために，読み込み，仕訳プログラム，サブルーチン，各種のテーブルなどを内部メモリー，コアメモリーに常駐させ，各種の文の処理プログラムを外部磁気ドラムに入れ，必要に応じてコア 300 語にブロック転送し処理する．

プログラムの大きさの制限は，テーブルの大きさで決定されるが HARP に比べかなり厳しい制限をつけている（**第 1 表**参照）．

第 1 表

変数名	100 以下
配列名	50 〃
文の番号	100 〃
〃（FORMAT 用）	50 〃
サブルーチン，関数副プログラム	30 〃

3.1 各種テーブル

プログラムを処理する段階で各種のテーブルが必要になるが，この処理プログラムでは以下のものを設けている．

（1） 変数テーブル

この計算機は 10 進 12 桁で，1 語の中に 6 文字入るが，HARP の変数名は 6 文字まで許されるので，変数の型，等式の左辺に現われたか右辺に現われたかの情報は別に格納している．

（2） 配列名テーブル

このテーブルには，配列名だけを格納している．元

数，型，使用ずみか否かの情報は格納してないため，多少問題がある．

（3） 文の番号テーブル

文の番号の情報は6カラム以前，以後に現われたか否かについてのみのせている．DO の入れ子の深さに関する情報は省略している．

（4） FORMAT テーブル

文の番号テーブルと同様．ただ入出力用と他のものを区別しているにすぎない．

（5） DO テーブル

DO 文で使用されている文の番号と制御変数を格納する．端末文が現われるごとに，対応する情報を消去する．

（6） 副プログラムテーブル

2種類あって一つはデックの中で CALL 文で呼ばれるサブルーチン名，算術式中に現われる関数副プログラム名の格納．他の一つは HARP 5020 で用意されている組込関数，基本外部関数の一覧表である．

3.2 チェックプログラム

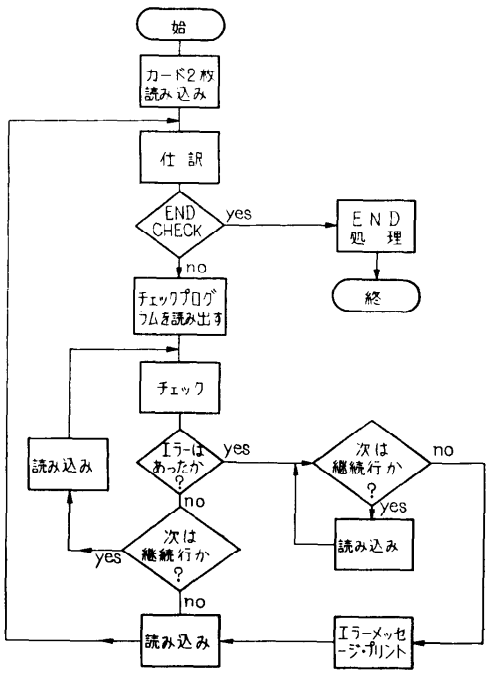
チェックプログラム全体の流れ図を**第1図**に示す．この処理プログラムでは常にカード2枚分のカードイメージを working storage I, II に格納しておく，それは1枚のカードの処理後次のカードの continuation mark の有無を調べる必要があるためである．ある文の処理中にエラーがあればその文のチェックを打切り次の文に移る．

読み込みルーチンは二つの部分から成っており，一つは紙テープを読み込みコード変換を行ない，カードイメージそのままに working storage II に格納する．他の一つはラインプリンタに working storage II の内容を印刷し，II から I へ FORMAT 文を除く総ての文の7カラム以降72カラムまでを空白を除き前につめて移す．実際のプログラムチェックの時は空白または72カラムまでチェックした場合そのカードのチェックは終りとする．FORMAT 文を除いたのは，Hollerith field があるためである．

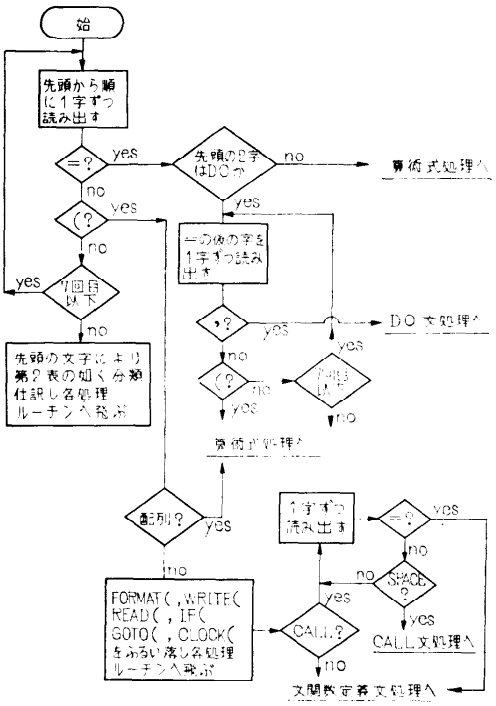
〔例〕 前　GObTObI, b(10, b 8, b 5)

　　後　GOTOI, (10, 8, 5)

なお working storage に常にカード2枚分が格納されているためと，END 文がないときのプログラムの暴走を防ぐために CHECK とせん孔したカードとブランクカードを最後に加えて紙テープに変換しておく．



第　1　図



第2図　仕訳流れ図

3.3 仕訳ルーチン

流れ図は**第2図**に示す．処理を行なわない文も仕訳だけは行なう．

第2表　仕訳の分類

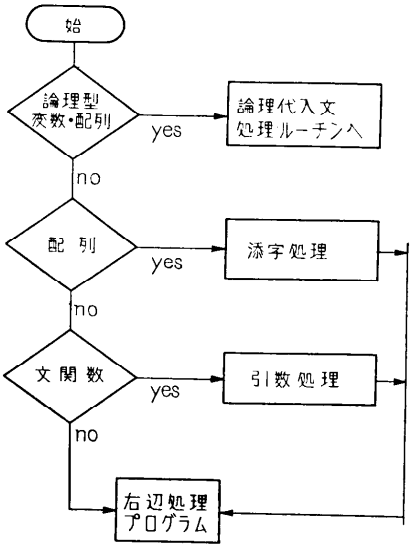
A	ASSIGN
B	BACK SPACE
C	COMMON, CALL, CONTINUE, COMPLEX, COMPLEX FUNCTION CHECK
D	DEBUG DUMP, DEBUG FLOW, DEBUG VALUES, DEBUG TERMINATE, DO, DOUBLE LENGTH INTEGER, DOUBLE LENGTH INTEGER FUNCTION, DOUBLE PRECISION FUNCTION, DOUBLE PRECISION COMPLEX, DOUBLE PRECISION, DOUBLE PRECISION COMPLEX FUNCTION, DRUM COMMON, DRUM DIMENSION, DIMENSION
E	END, END FILE, EXTERNAL, EQUIVALENCE
F	FUNCTION
G	GO TO
I	INTEGER, INTEGER FUNCTION
L	LOGICAL, LOGICAL FUNCTION
P	PUNCH, PAUSE, PRINT
Q	QUADRUPLE PRECISION, QUADRUPLE PRECISION FUNCTION
R	RETURN, REAL, REAL FUNCTION, READ DRUM, READ DRUM PROCEED, REWIND
S	STOP, SUBROUTINE
W	WORD LENGTH, WRITE DRUM, WRITE DRUM PROCEED

3.4 算術式，FORMAT，IF 文の処理

チェックプログラムは各文ごとに設けられているが代表的な例，算術式と FORMAT，それに IF 文の処理について述べる．

3.4.1 算術式処理

左辺処理と右辺処理とに分けている．左辺の処理で論理型であれば論理式処理ルーチンへ飛ぶ．他の場合は変数の型をのせて右へ処理ルーチンへ行く．ただし整数型のときには左辺の変数名が DO テーブルにある制御変数と一致するか否かを調べてから右辺処理に進む．右辺処理ルーチンでは compiler と同様演算子と演算数との棚を設けて処理を行なう．



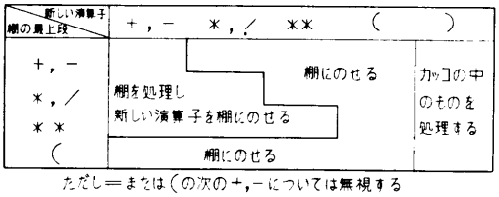
第3図　算術式左辺処理流れ図

演算数はタイプを次のように区別して棚にのせる．

（a）　整数型
（b）　実数型
（c）　複素数型
（d）　型の不明のもの（Function subprogram）

関数と配列の場合は引数と添字をそれぞれ処理したのち棚にのせる．棚の処理は**第3表**に従って行ない，その他演算数の組み合わせ，左辺と右辺の型の組み合わせはそれぞれ**第 4，5，6 表**に従い処理を行なう．

第　3　表



ただし＝または（の次の＋，－については無視する

第4表　演算数の組合わせ（×は許されない）

第一演算数 ＼ 第二演算数	＋　－　＊　／	整数型	実数型	複素数型
整数型				
実数型				
複素数型				

第5表　幅の場合の組合わせ（×は許されない）

＊　＊	整数型	実数型	複素数型
整数型			
実数型			
複素数型			

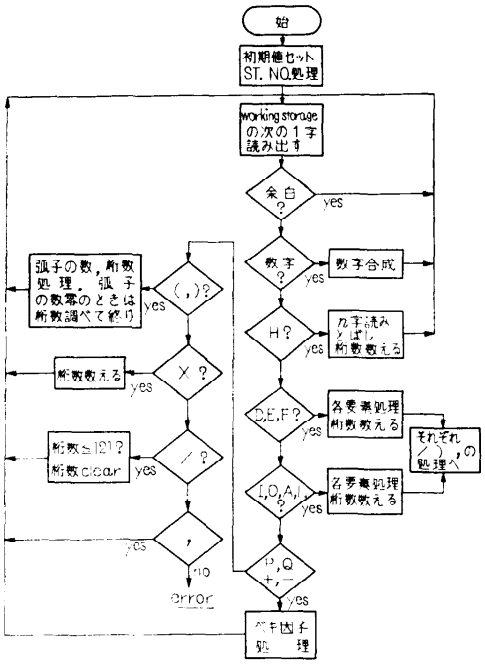
第6表　左辺・右辺の型の組合わせ
（×は許されない）

左辺 ＼ 右辺	整数型	実数型	複素数型
整数型			
実数型			
複素数型			

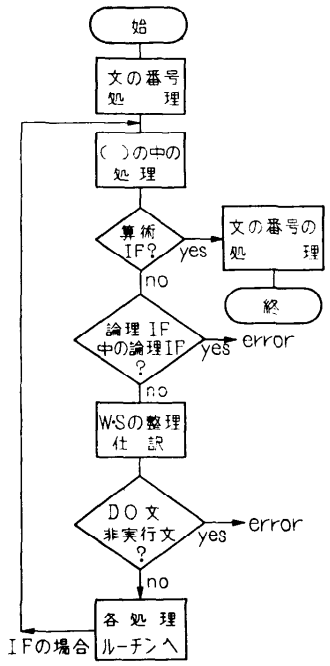
3.4.2 FORMAT 文処理

FORMAT 文は一番誤り易い文の一つであるため，compiler の過程で発見されるエラーはすべてチェッ

クし他に入力，出力の区別なく印刷桁数が 121 桁をこえるものはエラー表示をする．また D, E, F 変換を含む数値欄記述子については欄が w ケタを占め，そのうち小数部が d ケタとすると E, D 変換と F 変換の場合それぞれ $w \geq d+7$, $w \geq d+3$ をチェックしている．



第 4 図　FORMAT 処理ルーチン流れ図



第 5 図　IF 文処理の流れ図

FORMAT 文処理の流れ図を**第 4 図**に示す．

3.4.3　IF 文処理

第 5 図に流れ図を示す．図の中で w·s の整理とは仕訳ののち，各文の処理ルーチンへ飛ばすための準備で 1〜6 カラムを消し working storage I において IF（論理式）のあとの文を 7 カラム以降に格納しなおすことをいう．

3.5　END の処理

先に述べた各種テーブルを探すことにより次のことをチェックする．

（1）　変数名が左辺に現われ右辺に使われない．

（2）　変数の未定義

第 7 表　エラーメッセージ（その 1）

ERROR 1	MISSPELLING IN ST.
ERROR 2	NOT PERMISSIBLE IN HARP 5020
ERROR 3	ST. NO. IS NOT NUMERAL
ERROR 4	ST. NO. IS ZERO OR LARGER THAN 32767
ERROR 5	ST. NO. IS ALREADY DEFINED
ERROR 6	END ST. NOT APPEAR
ERROR 7	INVALID USAGE OF ST. NO.
ERROR 8	NOT TRANSFER NOR STOP ST. BEFORE END
ERROR 9	STOP OR RETURN ST. NOT APPEAR
ERROR 10	MORE THAN THREE DIMENSION
ERROR 11	FORM OF SUBSCRIPT IS INVALID
ERROR 12	VAR. IN SUBSCRIPT IS NOT INTEGER
ERROR 13	VAR. NAME IS MORE THAN 6 CHARACT.
ERROR 14	ALREADY USED IN OTHER TYPE
ERROR 15	FORM OF BRANCH NUMBER IS INVALID
ERROR 16	NO EXECUTABLE ST. IS ATTACHED
ERROR 17	DO ST. APPEAR IN IF ST.
ERROR 18	RELATIONAL EXPRESSION IS INVALID
ERROR 19	CALL NAME IS INVALID
ERROR 20	GO TO ST. IS INVALID
ERROR 21	LIST OF GO TO ST. IS INVALID
ERROR 22	TRANSFER ST. JUMPS INTO NO EXECUTABLE ST.
ERROR 23	DEVICE NO. IS NOT PERMISSIBLE
ERROR 24	FORMAT DESIGNATING ST. NO. IS INVALID
ERROR 25	FORMAT DESIGNATING VAR. IS USED WITH SUBSCRIPT
ERROR 26	LIST OF I/O ST. IS INVALID
ERROR 27	INDEXING VAR. IS NOT SIMPLE INTEGER
ERROR 28	FUNC. OR SUBR. ST. IS NOT FIRST ST.
ERROR 29	ILLEGAL SYMBOL IN FUNC. OR SUBR. ST.
ERROR 30	NO ARGUMENT IN FUNC. ST.
ERROR 31	DUMMY ARGUMENT APPEARS OVER AGAIN
ERROR 32	NO ST. NO. IN FORMAT ST.
ERROR 33	ILLEGAL KEY WORD OF FORMAT SPECI.
ERROR 34	EXCEED 1 BLOCK
ERROR 35	NUMERAL ELEMENT IN FORMAT IS INVALID
ERROR 36	FORM OF DO PARAMETER IS NOT PERMISSIBLE
ERROR 37	NEST OF DO LOOPS EXCEED 15
ERROR 38	INDEX OF DO ST. IS NOT INTEGER
ERROR 39	DO PARAMETER ZERO OR FIRST LARGER THAN SECOND
ERROR 40	INVALID NESTING OF DO LOOPS
ERROR 41	INDEX OF DO ST. IS CHANGED
ERROR 42	ARRAY VAR. IS DEFINED OVER AGAIN
ERROR 43	DIMENSION ST. IS INVALID
ERROR 44	ARRAY VAR. IS UNDEFINED
ERROR 45	ARITH. ST. IS INVALID
ERROR 46	DUMMY ARGUMENTS OF ST. FUNC. EXCEED 16
ERROR 47	ST. FUNC. IS NOT FIRST EXECUTABLE ST.
ERROR 48	RIGHT PARENTH. UNEQUAL LEFT ONES
ERROR 49	LOGICAL EXPRESSION IS INVALID
ERROR 50	RELATION OF BOTH SIDES IN ARITH. ST. IS NOT PERMISSIBLE

（3） 文の番号の未定義
（4） 文の番号が定義してあるが未使用である．
（5） DO の対応する端末文がない．
（6） 単純変数に添字がついている．
（7） 配列名であるのに添字がない．

6.7 については各文のチェック中に十分わかるものであるが，時間節約のため END の処理においてのみ検査する．上の処理を行なったのち，パンチミス発見を容易にするため次のテーブルを打ち出す．

（a） 変数テーブル
（b） 配列名テーブル
（c） 副プログラム名テーブル

4. エラーメッセージ

処理中に出すメッセージは**第7表**に，END の処理によるメッセージは**第8表**に示す．次にその簡単な例を示す．

5. おわりに

前にも述べたように，この計算機は非常に混んでいるため処理速度を重視したのとハードウェア上の制限からエラーの総数の8割のチェックを目標にしたが，いろいろの問題点を含んでいる．その一つは transfer statement と DO の入れ子の深さの関係を無視したために，DO の範囲のそとから DO の中への制御の移動をチェックしてないことである．

処理に要する時間はカード200枚のデックでオフラインのカード→紙テープ変換に約30分，プログラムチェックに約8分である．

（昭和42年4月6日受付）

第8表 エラーメッセージ（その2）

———	THIS VARIABLE IS UNDEFINED.
———	THIS VARIABLE AT LEFT SIDE IS NOT USED ANYWHERE.
———	THIS SIMPLE VARIABLE IS USED WITH SUBSCRIPTS.
———	THIS ARRAY VARIABLE IS USED AS SIMPLE VARIABLE.
———	THIS STATEMENT NUMBER IS UNDEFINED.
———	THIS STATEMENT IS NOT USED ANYWHERE.
NESTING	OF DO LOOPS IS INVALID.

```
$TEST       HARP                                              EFN  1
      DIMENSION  A(100)                                       EFN  2
      READ(5,100)  N                                          EFN  3
  100 FORMAT(I8)                                              EFN  4
      READ(5,101) (A(IH),IH=1,N+1)                            EFN  5
  101 FORMAT(5F12.0)                                          EFN  6
      READ(5,101) X0,DELTAX,XMAX                              EFN  7
      X=X0                                                    EFN  8
    1 P=A(1)                                                  EFN  9
      DO 10 IH=2,N+1                                          EFN 10

        ERROR 38  INDEX OF DO ST. IS NOT INTEGER

   10 P=P*X+A(IH)                                             EFN 11
      WRITE(6,103) X,P                                        EFN 12
  103 FORMAT(1H ,2HX=, E15.7,90X,2HP=,E15.7)                  EFN 13

        ERROR 34  EXCEED 1 BLOCK

      X=X+DELTEX                                              EFN 14
      IF(X.LE.XMAX) GO TO 1                                   EFN 15
      STOP                                                    EFN 16
      END                                                     EFN 17

   DELTAX  THIS VARIABLE AT LEFT SIDE IS NOT USED ANYWHERE
   DELTEX  THIS VARIABLE IS UNDEFINED

VARIABLE TABLE
     N      IH      X0  DELTAX    XMAX       X       P  DELTEX

ARRAY TABLE
     A

FUNCTION TABLE
```